UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/735,954	12/15/2003	Frederick S. Kaufmann	11223	3405

26890 7590 07/28/2009

JAMES M. STOVER
TERADATA CORPORATION
2835 MIAMI VILLAGE DRIVE
MIAMISBURG, OH 45342

EXAMINER	
STACE, BRENT S	
ART UNIT	**PAPER NUMBER**
2161	
MAIL DATE	**DELIVERY MODE**
07/28/2009	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE BOARD OF PATENT APPEALS
AND INTERFERENCES

Ex parte FREDERICK S. KAUFMANN,
GRACE KWAN-ON AU,
and VENKATA RAMAKRISHNA TIRUNAGARI

Appeal 2008-003140
Application 10/735,954[1]
Technology Center 2100

Decided:[2] July 28, 2009

Before JOHN C. MARTIN, LEE E. BARRETT, and
HOWARD B. BLANKENSHIP, *Administrative Patent Judges*.

BARRETT, *Administrative Patent Judge*.

DECISION ON APPEAL

This is a decision on appeal under 35 U.S.C. § 134(a) from the final rejection of claims 1-26. We have jurisdiction pursuant to 35 U.S.C. § 6(b).

We reverse.

[1] Filed December 15, 2003, titled "Row Triggers."

[2] The two-month time period for filing an appeal or commencing a civil action, as recited in 37 C.F.R. § 1.304, begins to run from the decided date shown on this page of the decision. The time period does not run from the Mail Date (paper delivery) or Notification Date (electronic delivery).

STATEMENT OF THE CASE

The invention

A method for processing a row trigger is described. The trigger is associated with a subject table in a relational database and defines a triggering statement and one or more triggered actions. *See* Abstract.

A "relational database" is a database that maintains a set of separate, related files (tables), but combines data elements from the files for queries and reports when required. The Specification describes:

> [C]hanges made to an entry in a particular table within the database may require changes to other entries. For example, a change to a product price entry may necessitate changes to a projected revenue entry. One method of implementing such changes is through the use of triggers.
>
> A trigger is a database object capable of executing specified triggered actions when a designated triggering statement occurs on a subject table with which the trigger is associated.

Spec. ¶¶ [0001]-[0002].

The Specification describes a distributed database system ("DBS") in connection with Figure 1. The DBS has one or more processing modules $110_{1...N}$ that manage the storage and retrieval of data in data storage facilities $120_{1...N}$. Spec. ¶ [0016]. The rows $125_{1...Z}$ of the tables are stored across multiple data-storage facilities $120_{1...N}$ to evenly distribute the workload. Spec. ¶ [0020]. The parsing engine 130 coordinates the retrieval of data from the data-storage facilities $120_{1...N}$ in response to queries received from a user at a mainframe 135 or a client computer 140. *Id.*

There are three important terms: triggering *statement*, triggering *action*, and transition table. "The triggering statement may include one of an UPDATE, INSERT, INSERT/SELECT, and DELETE statement to be executed on the subject table." Spec. ¶ [0005]. A triggering statement is converted into triggering statement instructions. Spec. ¶ [0027].
A transition table includes a transition table row, which includes first and second values associated with the subject table row. Spec. ¶ [0004]. "For example, the transition table may include the value of a particular row 125 before execution of the triggering statement and the value of the same row 125 after execution of the triggering statement." Spec. ¶ [0030].
Triggering action instructions are instructions to carry out operations on a another row of a table using the results in the transition table.

The method can be understood in connection with Figure 13, discussed in the Specification ¶¶ [0046]-[0053]. Note that the triggering action instruction sets in Figure 13 should be labeled 170a and 170b instead of 107a and 107b. Two tables, "Tab 1" having rows 125a and 125b, and "Tab 2" having rows 125x, 125y, and 125z, are shown in different data-storage facilities, 120_1 and 120_N, attached to different processing modules, 110_1 and 110_N, respectively. In the example code (¶ [0046]), which creates a "TRIGGER" after an "UPDATE," when a row in "Tab 1" is updated, the system will determine whether a row exists in "Tab 2" that contains the same value for "Emp_ID" and, if so, will delete the row in "Tab 2" and replace it with an updated version of the row from "Tab 1."

Parsing engine 130 receives an SQL request that includes an "UPDATE" statement. Parsing engine 130 determines that the "UPDATE" statement is a triggering statement for the row "TRIGGER" which, for example, affects rows 125a and 125b in "Tab 1" on data storage facility 120_1. As a result, parsing engine 130 identifies triggering statement processing module 110_1 based on its association with storage facility 120_1. Parsing engine 130 generates triggering statement instructions 150 that include an "UPDATE" instruction for each of rows 125a and 125b and generates triggered action instructions 170. The triggering statement instructions 150 and instructions to generate a transition table are transmitted to processing module 110_1, which updates rows 125a and 125b, and generates transition table 160. Transition table 160 includes transition table rows for both the old and new values of rows 125a and 125b (only old and new values for row 125a are shown). Parsing engine 130 receives the transition table 160, incorporates the values in the transition table 160 into sets of triggered action instructions 170a and 170b (mislabeled as 107a and 107b). Parsing engine 130 identifies a processing module or modules 110 corresponding to a data-storage facility that stores rows of "Tab 2" associated with the first set of triggered action instructions 170a and sends them to module 110_N. After processing the first set 170a, module 110N notifies parsing engine 130 and parsing engine 130 identifies a processing module or modules 110 corresponding to a data-storage facility that stores rows of "Tab 2" associated with the set triggered action instructions 170a and sends them to module 110_N. The processing is shown in Figures 14a-c.

An important part of the disclosed invention is the utilization of plural processing modules 110, each attached to a data-storage facility 120. "For a particular row trigger, multiple data-storage facilities 120 may contain rows of the subject table that are affected by the triggering statement and, thus, parsing engine 130 identifies multiple triggering statement processing modules." Spec. ¶ [0026]. That is, the parsing engine 130 must determine which processing module is associated with a subject table row that is affected by a triggering statement and which processing module is associated with a row that is affected by the triggering action.

Illustrative claim

Claim 1 is reproduced below:

> 1. A method for processing a trigger associated with a subject table in a relational database, wherein the trigger defines a triggering statement and one or more triggered actions, the method including:
>
> determining that a triggering statement of a trigger will execute on a subject table row of a subject table;
>
> requesting a transition table in response to determining that the triggering statement will execute, the transition table including a transition table row, wherein the transition table row comprises at least one value associated with the subject table row;
>
> reading the transition table row from the transition table;
>
> identifying a processing unit to receive the transition table row and a triggered action of the trigger based on an association between the identified processing unit and a portion of memory; and
>
> transmitting the transition table row and the triggered action to the identified processing unit to be processed.

The references

Chen	US 5,930,795	Jul. 27, 1999

"Database trigger," http://en.wikipedia.org/wiki/Database_Trigger, (last visited June 27, 2007), 3 pages ("Wikipedia").

"Transition tables," IBM, http://publib.boulder.ibm.com/infocenter/db2luw/v8/index.jsp?topic=/com.ibm.db2.udb.doc/ad/c0006671.htm (last visited Oct. 31, 2006) ("IBM").

David J. DeWitt and Jim Gray, *Parallel Database Systems: The Future of High Performance Database Processing*, footnote 1 states that the article appeared in Communications of the ACM, Vol. 35, No. 6, June 1992 ("DeWitt").

The rejection

Claims 1-26 stand rejected under 35 U.S.C. § 102(b) as being anticipated by Chen. The rejection cites to the discussion of "database trigger" and "row trigger" in Wikipedia, the discussion of "transition tables" in IBM, and the discussion of parallel database systems in DeWitt.

The Examiner refers to the following portions of Chen for the limitations of claim 1: column 5, lines 50-54, 60-67; column 12, lines 25-67; column 13, lines 1-5; and column 14, lines 8-50. Final Rej. 8-9; Ans. 4-5. The Examiner states that "in a parallel processing environment, the different processors process different activities; for a particular job a processor has to be identified to perform the job." Final Rej. 9.

The Examiner rearranges the rationale of the Final Rejection in the Answer and we mostly refer to this later version as the Examiner's position, both as to the statement of the rejection and the response to the arguments.

The Examiner notes that IBM defines a "transition table" as a table that captures the value that is used to update the rows in a database when the triggered action is applied in the database. Ans. 6-7. The Examiner notes that Wikipedia defines a "database trigger" as procedural code that is automatically executed in response to certain events on a particular table in a database and a "row trigger" as a class of database trigger that defines an action for every row of a table. *Id.* at 7. The Examiner cites DeWitt to show that in a parallel database system architecture, processors communicate only by sending messages and "tuples of each relation in the database are partitioned (declustered) across disk storage units attached directly to each processor." *Id.*

The Examiner finds that Chen is applicable to any type of database management system including those on a parallel processing system, noting column 5, lines 49-54. Ans. 8. The Examiner impliedly finds that a parallel processing database management system has a subject table on a data storage associated with a first processor which receives a triggering statement and a table having data that will be affected by the trigger on a different data storage associated with a second processor. It is stated:

> Therefore, in order to execute properly, the database management system identifies that particular second processor, since it has the data, and sends the transition table data to that processor to be processed. Such characteristics is [sic] fundamental to any functioning distributed

> parallel database system . . . . Examiner is not sure what the Appellant is claiming since the Appellant appears to be claiming a fundamental characteristic of a database system, which has existed for decades . . . .

Ans. 9.

The Examiner finds that in a parallel processing database management system, "[t]he table(s) and/or row(s) are partitioned across multiple disks, wherein the disk(s) are attached directly to the processors." Ans. 9. The Examiner states:

> Therefore *if* Table AB row is divided into disk A (and attached processor A) and disk B (and attached processor B) and an UPDATE on the table AB row triggers an event, the transition row would capture the new values and a processor *has to be identified* (i.e., a disk/memory portion and the associated processor has to be identified, which stores the relevant portion of the table row that needs to be updated) to process the UPDATE (if a disk does not have the portion of the table that needs to be updated there is no point in going to that disk).

Final Rej. 5 (emphasis added).

The Examiner finds:

> All of the "receiving triggered statement", "determining triggered statement", "computer program", "executing triggered statement" are all *inherent* characteristics of any database management system supporting triggers. Without receiving, determining and executing, a triggered statement cannot be processed. Identifying a processor to process an event (e.g., UPDATE, DELETE etc.) is *inherent* to a distributed parallel database system, where data is distributed/partitioned among different portions of memory (e.g., different disks), because simply a processor must be used to process commands/events and there is no need to access other portions of

> memory/processors when the other memory portions do not have the corresponding data that needs to be processed.

Ans. 10 (emphasis added).

The Examiner finds that the limitations of the rest of the claims are covered by the analysis of claim 1 (Ans. 5).

PRINCIPLES OF LAW

An examiner bears the initial burden of establishing anticipation. Once the examiner finds a prima facie case of anticipation based on a prior art reference, the burden shifts to the patent applicant to show that prior art reference does not inherently possess the claim limitations of claimed apparatus. *In re Schreiber*, 128 F.3d 1473, 1478 (Fed. Cir. 1997).

"[U]nless a reference discloses within the four corners of the document not only all of the limitations claimed but also all of the limitations arranged or combined in the same way as recited in the claim, it cannot be said to prove prior invention of the thing claimed and, thus, cannot anticipate under 35 U.S.C. § 102." *Net MoneyIN, Inc. v. VeriSign, Inc.*, 545 F.3d 1359, 1371 (Fed. Cir. 2008).

"Inherency, however, may not be established by probabilities or possibilities. The mere fact that a certain thing may result from a given set of circumstance is not sufficient." *In re Oelrich*, 666 F.2d 578, 581 (1981) (quoting *Hansgirg v. Kemmer*, 102 F.2d 212, 214 (CCPA 1939)).

Extrinsic evidence may be considered to demonstrate that missing descriptive matter is necessarily present (inherent) to a person of ordinary skill in the art, but not to add matter to fill the gaps:

> To serve as an anticipation when the reference is silent about the asserted inherent characteristic, such gap in the reference may be filled with recourse to extrinsic evidence. Such evidence must make clear that the missing descriptive matter is necessarily present in the thing described in the reference, and that it would be so recognized by persons of ordinary skill.

Metabolite Laboratories, Inc. v. Laboratory Corp. of America Holdings, 370 F.3d 1354, 1368 (Fed. Cir. 2004) (quoting *Continental Can Co. v. Monsanto Co.*, 948 F.2d 1264, 1268 (Fed. Cir. 1991)). *See also Scripps Clinic & Research Foundation v. Genentech, Inc.*, 927 F.2d 1565, 1576 (Fed. Cir. 1991) ("The role of extrinsic evidence is to educate the decision-maker to what the reference meant to persons of ordinary skill in the field of the invention, not to fill gaps in the reference.").

FINDINGS OF FACT

Chen relates to a database management system for processing SQL queries for relational databases, and more specifically to compiling the queries and processing the queries at run-time where the queries reference a table entity that is unknown at compile time, i.e., providing dynamic table linking at run-time. Col. 1, ll. 15-19.

Chen describes that the method is "applicable to any type of database management system whether it is contained within a single system or is

within a networked environment including parallel processing systems, client/server processing systems, distributed systems, etc." Col. 5, ll. 50-54.

Chen describes "external" transition tables, i.e., tables that cannot be identified at compile-time, and dynamic linking of transition table references appearing in an external trigger body. Col. 12, ll. 1-65. "Transition tables are temporary tables that capture the state of affected rows when the triggering SQL operation is applied to a table." Col. 12, ll. 2-4.

Chen describes generating a transition table which is populated by the value of the affected rows. Col. 12, ll. 25-65.

Chen provides an example of a row trigger which processes an old and new transition table. Col. 14, ll. 8-51.

ISSUE

Has the Examiner carried the initial burden of showing that Chen, as implemented in a parallel database system, inherently includes separate processing units for creating a transition table in response to a triggering statement and for receiving the transition table and the triggered action or inherently identifies a particular processing unit that is to receive the transition table and the triggered action, so as to shift the burden of going forward with the evidence to Appellants?

CONTENTIONS

Appellants argue that "the Examiner has merely concluded that elements of a number of different claims are an inherent part of the operation of Chen without considering the possibility that alternatives to the relevant

claim elements might be used in, what the Examiner labels as, 'process[ing]' a triggered statement." Br. 7.

Appellants argue that "these allegedly inherent properties only result from a hypothetical situation that the Examiner has himself crafted." Br. 7. Appellants point to the Examiner's example that "*if* Table AB row is divided into disk A . . . and disk B . . . and an UPDATE on the table AB row triggers an event, the transition row would capture the new values and a processor *has to be identified*" (Final Rej. 5), which is clearly conditional. Br. 8.

Appellants argue that the Examiner has provided no evidence to support assertions about the operation of the database system. Br. 8.

As to claims 1-3, 11-13, and 21-26, Appellants argue that Chen fails to disclose "identifying a processing unit to receive the transition table row and a triggered action of the trigger based on an association between the identified processing unit and a portion of memory." It is argued that of the portions of Chen cited by the Examiner, "[n]one of these cited portions address identifying a processing unit to receive the transition table row and a triggered action of the trigger based on an association between the processing unit and a portion of memory, as required by claim 1." Br. 9.

As to claims 4-6 and 14-16, Appellants argue that "the Examiner fails to specifically address the individual limitations of Claim 4, instead basing the rejection of Claim 4 on the language of Claim 1." Br. 9. Appellants argue that Chen fails to disclose, at least, "instructing a first processing unit, in response to determining that the triggering statement of the trigger will execute, to communicate a transition table row to a second processing unit,

wherein the transition table row comprises at least one value associated with the subject table row" as recited by claim 4. *Id.* at 9-10. It is stated that claim 14, although of differing scope from claim 4, includes elements that, for reasons substantially similar to those discussed with respect to claim 4, are not disclosed by Chen. *Id.* at 10.

As to claims 7, 8, 17, and 18, Appellants argue that "the Examiner also fails to specifically address the individual limitations of Claim 7." Br. 10. It is argued that Chen fails to disclose "receiving a triggering statement of a trigger to be executed on a subject table row of a subject table and *information identifying a processing unit*" (emphasis added) as recited by Claim 7. *Id.* Appellants state that claim 17, although of differing scope from claim 7, includes elements that, for reasons substantially similar to those discussed with respect to claim 7, are not disclosed by Chen. *Id.*.

As to claims 9, 10, 19, and 20, Appellants argue that "the Examiner also fails to address the individual limitations of Claim 9." Br. 10. It is argued that Chen fails to disclose "receiving a triggered action of a trigger associated with a subject table and <u>*information identifying the transition table row*</u>" (emphasis by Appellants) as recited by claim 9. *Id.* Appellants argue that claim 19, although of differing scope from claim 9, includes elements that, for reasons substantially similar to those discussed with respect to claim 9, are not disclosed by Chen. *Id.*

Appellants argue that the Examiner errs in asserting that simply because a transition table can exist in a parallel processing system, the elements of claim 1 are inherent. Reply Br. 2. It is argued that Chen does

not teach "reading a row from the transition table, identifying a processing unit to receive the row and a triggered action, and transmitting the row to the identified processing unit" *Id.* It is argued that Chen treats transition tables as a whole and does not discuss processing individual rows of the transition table, so "[i]f Chen were used as a guide in determining how a parallel processing system would process transition tables, the entire transition table, not just a row, would be transmitted to a processing unit." *Id.* at 3.

ANALYSIS

Initially, we find that Chen teaches, at columns 12 and 14, a triggering statement that will execute on a row of a subject table in a relational database, creating a transition table including at least one value associated with the affected row of the subject table, and processing the transition table in accordance with triggering action instructions. That is, there appears to be nothing novel in the relational database art about triggering statements, transition tables, and triggering actions and their operation. Appellants do not argue that they have invented triggering statements, transition tables, and triggering actions. The issue of patentability involves triggering in a system implicitly or explicitly having plural processing units.

As a matter of claim interpretation, except for independent claims 4 and 14, which expressly recite a first processing unit communicating a transition table row to a second processing unit, which processes a received triggered action based on the transition table row, we interpret the independent claims to implicitly require separate processing units for

creating a transition table in response to a triggering statement and for receiving the transition table and the triggered action, with some claims further requiring identification of the processing unit that is to receive the transition table and the triggered action. For example, claim 1 recites "transmitting the transition table row and the triggered action to the identified processing unit to be processed," which implicitly requires that there is a processing unit transmitting the transition row table and triggered actions to an "identified processing unit"; claims 11 and 21 are similar. Claim 7 recites "transmitting the transition table row to the processing unit," which also implies a processing unit to transmit to another processing unit; claim 17 is similar. Claim 9 recites "receiving a transition table row" and "receiving a triggered action of a trigger associated with a subject table," which implicitly requires that the transition table row and triggered action come from different processing units; claim 19 is similar.

The Examiner finds that Chen is applicable to any type of database management system including those on a parallel processing system, referring to column 5, lines 49-54. Ans. 8. The Examiner refers to DeWitt for a description of distributed parallel database system architectures and finds that DeWitt teaches that in a "shared-nothing" design, processors communicate only by sending messages, each processor has its own memory, and "tuples of each relation in the database are partitioned (declustered) across disk storage units attached directly to each processor." Ans. 7. The Examiner impliedly finds that a parallel processing database management system has a subject table on a data storage associated with a

first processor which receives a triggering statement and a table having data that will be affected by the trigger on a different data storage associated with a second processor. The Examiner finds that the system must (inherently) identify the processor that generates and transmits the transition table row and must (inherently) identify the processor that receives the transition table row and the triggered actions because "[s]uch characteristics is [sic] fundamental to any functioning distributed parallel database system" (Ans. 9).

Appellants' argument that Chen treats transition tables as a whole and does not discuss processing individual rows of the transition table (Reply Br. 2-3) is not persuasive because the claims do not preclude more than one transition table row.

DeWitt teaches several different implementations of parallel database systems. Chen does not mention a particular type of parallel design, so in order for the Examiner's anticipation rejection based on inherency to be successful, it must be that all implementations necessarily operate as claimed. "Inherency, however, may not be established by probabilities or possibilities. The mere fact that a certain thing may result from a given set circumstance is not sufficient." *Oelrich*, 666 F.2d at 581.

DeWitt describes parallel database systems. The Examiner does not establish that the claims cover both kinds of parallel systems (pipeline parallelism or partitioned parallelism) shown in DeWitt, Figure 1. In fact, claim 1, for example, recites transmitting a transition table row and a triggered action to an *identified* processing unit, which does not even appear

to be a parallel processing operation where operations are conducted simultaneously.

DeWitt describes several different hardware architectures: a "shared-nothing" design where each processor has its own memory and disk; a shared memory multiprocessor; and a shared disk multiprocessor. Figs. 3 and 4, p. 7. The Examiner has not shown that all of these different parallel architectures necessarily perform the transmitting and receiving steps claimed. The Examiner refers to the "shared-nothing" design, but Chen is not limited to a particular type of parallel design. It has not been shown that the shared memory and shared disk systems meet the transferring limitation. The fact that one of the architectures *may possibly* operate as claimed does not prove inherency.

In addition, the claims depend on where the rows of the subject table and the rows affected by the trigger action are located, not just on the hardware architecture. DeWitt describes that data can be partitioned, where "[p]artitioning a relation involves distributing its tuples over several disks" (p. 11); a "tuple" corresponds to a "row" (Chen, col. 1, ll. 26-27). However, the Examiner has not established that data needs to be or is always partitioned this way. For example, the Examiner has not established that all the activity for a relational database cannot take place on one processor, in which case there would be no inherency. We agree with the Examiner that it is logical that a relational database distributed over several processors, each having its own memory, would necessarily have to identify which processor executes the triggering statement instructions and have to identify which

processor to send the transition table row for executing the triggering action, but the Examiner has not provided sufficient evidence that this must inherently happen in all parallel databases. Our reviewing court, the U.S. Court of Appeals for the Federal Circuit, requires facts, not speculation.

The Examiner states that "Examiner is not sure what the Appellant is claiming since the Appellant appears to be claiming a fundamental characteristic of a database system, which has existed for decades." Ans. 9. If true, it should be easy to provide an express teaching rather than relying on inherency based on a mention of "parallel" database systems.

We find that the Examiner has failed to establish that implementing row triggers in a parallel database system inherently requires transmitting a transition table row to an identified processor as claimed. Accordingly, the burden of going forward with the evidence has not shifted to Appellants to show that prior art reference does not inherently possess the claim limitations of claimed apparatus. *Schreiber*, 128 F.3d at 1478.

CONCLUSION

The Examiner has not carried the initial burden of showing that Chen, as implemented in a parallel database system, inherently includes separate processing units for creating a transition table in response to a triggering statement and for receiving the transition table and the triggered action or inherently transmits a transition table row to an identified processor, so as to shift the burden of going forward with the evidence to Appellants.

The rejection of claims 1-26 under 35 U.S.C. § 102(b) over Chen is reversed.

REVERSED

rwk

JAMES M. STOVER
TERADATA CORPORATION
2835 MIAMI VILLAGE DRIVE
MIAMISBURG, OH 45342